SYSTEM AND METHOD FOR MONITORING AND ANALYZING AGREEMENTS BETWEEN PARTIES

Cross-Reference to Related Applications

This application claims the benefit of U.S. Provisional Application Serial No. 60/454,919 filed on March 14, 2003, which is incorporated herein by reference.

Field of the Invention

The present invention generally relates to enterprise software, and more particularly, to a system for monitoring and analyzing agreements between two parties, which is operative to calculate acceptable levels of supply based on contractual obligations and forecasted demand.

Background of the Invention

In many customer and supplier relationships, an agreement is put into place in order to meet certain objectives of each party. Particularly, agreements may be executed to obligate a customer to reimburse the supplier for that investment if the customer does not ultimately purchase the item. To ensure a fair settlement, an agreement may define under what conditions the supplier's investment in the un-purchased item must be reimbursed by the customer to the supplier. An agreement may also obligate the supplier to pay a penalty to the customer for any under-investment made which results in the supplier being unable to successfully fulfill when the customer ultimately purchases the item.

It is often difficult to determine whether the both parties' objectives are being met and to measure and monitor the overall performance under an agreement. Some parties have implemented systems to try to solve this business problem. However, none of these prior systems provide the necessary data or can be easily modified to deliver the desired outputs. Some parties may also use informal, ad-hoc methods to monitor and measure the performance under an agreement. Such methods are unreliable and provide poor and inconsistent results.

It would therefore be desirable to provide a system for defining, measuring, and monitoring agreements between two parties to a contract, which will allow the parties to consistently and objectively calculate acceptable levels of supply based on contractual obligations and forecasted demand.

Summary of the Invention

The present invention provides a system for monitoring and analyzing agreements between parties. The system calculates the quantity of inventory and orders for an item that should be in place based on previous forecasted demand for that item. This is critical information between a customer and a supplier because suppliers are frequently forced to make a financial investment in creating an item that is sold to a customer before that customer actually buys that item. In many customer and supplier relationships, an agreement is made or executed that accomplishes two things:

1. It obligates that customer to reimburse the supplier for that investment if the customer does not ultimately purchase the item. To ensure a fair settlement, the contract defines under what conditions the supplier's investment in the un-purchased item must be reimbursed by the customer to the supplier.
2. It obligates the supplier to pay a penalty to the customer for any under-investment made which results in the supplier being unable to successfully fulfill when the customer ultimately purchases the item.

The present system defines two concepts to help determine the ongoing obligation between a supplier and a customer. First, the resulting amounts calculated for these concepts are defined as "liability" from one party to the other. And second, these amounts are determined by a "good faith" algorithm which tracks past performance against contractual commitments to determine liability. As stated above, the liability can be calculated in two conditions: (i) where a supplier has made an investment for which a customer must reimburse the supplier; and (ii) where the supplier has not made enough investment and therefore is penalized by the customer.

An example of the first condition can be seen in an outsourced manufacturing relationship between the company that sells a final assembly in the marketplace (the customer) and a contract manufacturer that procures the material to assemble that assembly and assembles that assembly (the supplier). Figure 1A illustrates such a condition. Because the lead times to procure the material are likely much longer (several weeks) than the lead time for the procurement of the final assembly (several days) between the customer and supplier, the supplier is forced to make a financial investment in inventory and pending orders for that material before the customer has made a firm order for the final assembly. The supplier makes decisions on how much investment to make based on how much the customer has indicated that it is planning to buy (forecasted demand). Even though this forecasted demand is not a firm order to buy the assembly, most of these customer and supplier relationships are governed by a contract that defines what happens when the customer does not buy as much as the forecast. Frequently, this contract will state that the customer is obligated to reimburse the supplier for the investment made in specific material (not all), which was driven by the forecast and the lead-time for that material.

An example of the second condition can be shown using the same relationship, where the supplier makes a mistake or for strategic reasons chooses to not invest enough to meet the forecast provided by the customer. Figure 1B illustrates such a condition. In this condition, the lead times for the material to make the assembly are significantly longer than the committed order time for the assembly between the customer and supplier, and thus it becomes either physically impossible or possible only at a significantly higher investment for the supplier to meet the actual purchase of the item by the customer. In this case, the contract will state how much the supplier has to pay the customer in penalty or that the supplier cannot pass the increased investment required to the customer.

In the first condition, the supplier is financially motivated to maximize the amount of the reimbursement and the customer is financially motivated to minimize the amount of the reimbursement; and in the second condition, the supplier is financially motivated to minimize the penalty and the customer is motivated to maximize the penalty although the customer does not have complete information. As a result, the quantifying of the investment is a potentially

contentious process. Existing systems and methodologies are not capable of this calculation.

The present system is adapted to keep track of this information and to calculate the obligations each party has to the other based on the contract that exists between them and how they have performed relative to the stipulations in that contract. The information required for such a system not only includes the actual forecasts and performance, but also additionally includes when that information was shared between parties. Without the "when it was shared" component, it would not be possible to calculate how much investment should be made at that time or to calculate how much over or under investment has been made relative to that amount. Figure 1C illustrates the calculations and input and output capabilities provided by the present invention.

Presently, there are no known systems or algorithms that provide these calculations. Some systems do exist, which people have tried to use to solve some of these business problems. However, none of these systems provides the necessary data or calculations, nor can be readily modified to deliver the desired outputs.

Some conventional systems that either (or both) a customer or supplier may use include:

1. Material Requirements Planning systems
2. Advanced Planning Systems
3. Contract Management Systems
4. Data warehouse/Business Intelligence systems

Materials Requirements Planning (MRP) systems

Customers and suppliers may have systems that help them decided how much investment in materials needs to be made at a present date so that a certain level of final assemblies can be made at a future date (e.g., materials requirements planning systems (MRP)). Figure 1D illustrates a conventional MRP system. The MRP system is capable of taking in a present known situation and determining what investment is required going forward. However, it falls short in its capability to deliver the acceptable levels of investment and the potential customer reimbursement or supplier penalty because it lacks at least three components:

1. It does not take into account any modifications that the contract may make to the customer financial responsibilities.
2. It does not take into account when the information was shared when determining who is responsible for a reimbursement or penalty. It only knows how much excess exists or how much cannot be built.
3. It plans future investment without identifying what the effect of past mistakes and excesses are.

Advanced Planning Systems (APS)

It is possible that both the customer and supplier will have advanced planning systems that help them optimize when their investment needs to be made, and to minimize that investment while maximizing some other aspect like revenue or fulfillment rate. Figure 1E illustrates a conventional advanced planning system (APS). Like the MRP system, the APS system is capable of taking in a present known situation and determining what investment is required going forward. However, it falls short in its capability to deliver the acceptable levels of investment and the potential customer reimbursement or supplier penalty because it lacks at least four components:

1. It does not take into account specific contractual obligations for reimbursement or penalties.
2. It does not take into account when the information was shared when determining who is responsible for a reimbursement or penalty. It only knows how much excess exists or how much cannot be built.
3. It plans future investment without identifying the effect of past mistakes and excesses.
4. It does not store historical data which is auditable and usable to calculate current acceptable levels.

Contract Management Systems

It is possible that both the customer and supplier will have systems that help them capture the details of their contractual obligations in an electronic form. Figure 1F illustrates a typical contract management system. These contract management systems may be capable of documenting the agreed upon methods and amounts that reimbursements will be calculated.

However, most fall short in their inability to monitor actual performance against those obligations and, therefore, cannot calculate the result of current performance. While there are some existing contract management systems that do track actual performance against contractual obligations, those systems do not propagate an order or forecast for an item into the required investment in materials to ultimately deliver that item. Those systems only track the commitment at the item level itself. For example,

1. they do not track actual supplier investment;
2. they do not calculate what acceptable additional investment must be made to support future forecasts and orders; and
3. they do not calculate customer reimbursement or supplier penalty based on historical information.

Data warehouse/Business Intelligence Systems

It is possible that both the customer and supplier will have data warehouse and/or business intelligence systems that help them capture historical information on their performance as well as those of their suppliers or customers. Figure 1G illustrates a conventional data warehouse/business development system. Additionally, it is possible that these systems also capture the details of the contractual terms that are in place with their customers and suppliers. These systems, though, are solely a repository of data and do not have the required algorithms in place to calculate the acceptable levels of supply based on contractual obligations and forecasted demand. For example,

1. they do not identify required investment to meet future forecasts and orders;
2. they do not segment current performance by what is acceptable and what is not; and
3. they do not calculate auditable reimbursements or penalties.

The present invention provides a system for defining, measuring, and monitoring agreements between two parties, which overcomes all of the foregoing drawbacks and limitations of these prior systems, and which accurately, objectively and consistently calculates acceptable levels of supply based on contractual obligations and forecasted demand.

According to one aspect of the invention, a system is provided for analyzing agreements between a first party and a second party. The system is adapted to receive input data describing commitments and past performance between the first and second party, and to calculate a liability of the first and second party to one another based on the data.

The input data may include contractual terms between the first and second party forecasted demand from the first party, actual demand from the first party, and/or actual investment from the second party. The liability may include an investment required by the second party to meet the forecasted demand, reimbursement to the second party by the first party for over investment, and/or a penalty from the second party to the first party for under investment.

According to another aspect of the invention, a computerized method is provided for analyzing agreements between a first party and a second party. The computerized method includes receiving input data describing commitments and past performance between the first and second party; and calculating liability of the first and second party relative to each other based on the data.

According to another aspect of the invention, a computer-readable medium is provided having computer-executable instructions for performing a method for analyzing agreements between a first party and a second party. The method includes the steps of: receiving input data describing commitments and past performance between the first and second party; and calculating liability of the first and second party relative to each other based on the data.

These and other features and advantages of the invention will become apparent by reference to the following specification and by reference to the following drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A is a block diagram illustrating a first condition in an outsourced manufacturing relationship between a company that sells a final assembly in the marketplace and a contract

manufacturer that procures the material to assemble that assembly and assembles that assembly.

Figure 1B is a block diagram illustrating a second condition, where a supplier makes a mistake or for strategic reasons chooses to not invest enough to meet the forecast provided by a customer.

Figure 1C is a block diagram illustrating the input and output capabilities provided by the present invention.

Figure 1D is a block diagram illustrating a conventional MRP system.

Figure 1E is a block diagram illustrating a conventional advanced planning system (APS).

Figure 1F is a block diagram illustrating a conventional contract management system.

Figure 1G is a block diagram illustrating a conventional data warehouse/business development system.

Figure 2 illustrates one embodiment of an enterprise system for monitoring and analyzing agreements between parties, according to the present invention.

Figure 3 is a block diagram illustrating the inputs and outputs of the good faith calculation that may be performed by the present invention.

Figures 4 and 5 are block diagrams illustrating simple examples of extended supply chains.

Figure 6 is a block diagram illustrating an example of a simple Bill of Materials (BOM).

Figure 7 is a block diagram showing a top level of the good faith calculation.

Figure 8 is a flow diagram showing a process of propagating the demand for an item, which may be employed by the present invention.

Figures 9A and 9B are flow diagrams showing a process of getting the independent demand for an item, which may be employed by the present invention.

Figure 10 is a flow diagram showing a process for evaluating each item, which may be employed by the present invention.

Figure 11 is a flow diagram showing a process for tracking and updating the disposition of supply at a site with respect to good faith assessment of validity, which may be employed by the present invention.

Figure 12 is a flow diagram showing a process for tracking changes to the status of orders for an item, which may be employed by the present invention.

Figure 13 is a flow diagram showing a process for adjusting the good faith status of the inventory to match the quantity actually loaded from an Enterprise Resource Planning (ERP) system, which may be employed by the present invention.

Figure 14 is a flow diagram showing a process for determining which inventory is consumed.

Figure 15 shows a process for promoting inventory from invalid to valid, which may be employed by the present invention.

Figure 16 is a flow diagram showing a process for promoting orders from invalid to valid, which may be employed by the present invention.

Figure 17 is a flow diagram showing a process for the calculation of the liability terms, which may be employed by the present invention.

Figure 18 depicts the results of aligning two Time Series.

Figure 19 is an example of a user interface that the system may generate to display the names and description of certain terms that may be present in an agreement.

Figure 20 is an example of a user interface that the system may generate to display parameters and calculations that are associated with a term.

Figure 21 is an example of a user interface that the system may generate to display Liability that exists between a company and a customer.

Figure 22 is an example of a user interface that the system may generate to display the good faith results calculated for an item in a certain period.

Figure 23 is an example of a user interface that the system may generate to display the calculated inventory and order levels for a specific item base on the agreement terms and the forecasts, order and inventory that exist for that item.

DETAILED DESCRIPTION OF THE EMBODIMENTS

The present invention will now be described in detail with reference to the drawings, which are provided as illustrative examples of the invention so as to enable those skilled in the art to practice the invention. Notably, the implementation of certain elements of the present invention may be accomplished using software, hardware, firmware or any combination thereof, as would be apparent to those of ordinary skill in the art, and the figures and examples below are not meant to limit the scope of the present invention. Moreover, where certain elements of the present invention can be partially or fully implemented using known components, only those portions of such known components that are necessary for an understanding of the present invention will be described, and detailed descriptions of other portions of such known components will be omitted so as not to obscure the invention.

Figure 2 illustrates one embodiment of an enterprise system 10 with a rich set of features for monitoring and analyzing agreements between parties, according to the present invention.

The system 10 is adapted to calculate acceptable levels of supply based on contractual obligations and forecasted demand. For the purposes of explaining the system, the following elements depicted in Figure 2 will be described:

1. Outsourced manufacturing data model 30
2. Agreement management module and terms library 40
3. Liability calculation engine 50
4. Good Faith calculation engine 60

It should be appreciated that each of the portions or blocks illustrated in Figure 2 (as well as the portions or blocks illustrated in the other Figures) may represent the hardware and/or software utilized to perform the described calculations, steps and processes. In the preferred embodiment, any one or more of the portions or blocks shown may be implemented in a computer readable medium. Furthermore, it should be appreciated that the processes described below are not limited to the specific steps or functional blocks that are shown, but that additional or different steps may be included and that the steps may be performed in any suitable order.

The system 10 may be implemented using a conventional and commercially available computing system. The computing system may include a processing and memory unit, having a microprocessor, volatile and non-volatile memory, and one or more persistent storage devices. In the preferred embodiment, the processing and memory unit may be adapted to store and run at least a portion of the operating software which directs the operation of the system. The computing system may also include one or more input devices for providing data to and accessing data from the system, such as a keyboard, mouse, and other conventional peripheral devices such as disk drives, printers, scanners and the like. An output or display unit (e.g., a computer monitor, a flat panel display, printer or other display device) may be used to graphically display data to a user. The input devices and display unit cooperatively permit a system user to operate the system, input data into the system, and access data and results from the system. The computing system may further include a communications unit for transferring data over a network, such as a local or global computer network, or a wireless network.

Referring back to Figure 2, the outsourced manufacturing data model 30 is a computer model that understands typical transactions and relationships that occur within and across trading partners in a supply chain. The data model 30 includes relational databases containing information describing items and bills of materials (BOM) 12, trading partners 14 and transactions 16. Most of the transactions 16 can be found in conventional ERP systems, which may interface with system 10. The data model 30 goes beyond ERP through understanding the translation of information from one trading partner to another. For example, the system 10 understands that an item called "ABC" at one trading partner may be called "XYZ" at another. This item mapping allows users to tie transactions created in one item number to those to transactions in a trading partner's system. For example, a purchase order created in a customer's system is related to a work order and inventory in the supplier's system through the mapped item numbers.

The agreement management module 40 allows the user to set up agreements between trading partners. The agreement management system includes relational databases describing predefined terms 18 and trading partner agreements (TPAs) 20. As used herein, TPAs should be understood to represent an embodiment of the commitments made between a customer and a supplier (or vendor). As used herein the term "agreement" should be understood to include formal contracts, informal agreements and other informal relationships that may between parties. Because of the buy-sell relationship between a customer and supplier, each party is known as a trading partner. The commitments made between trading partners range from the pricing of products to be bought, the quantities of product to be bought, the manner in which product will be bought, the method in which payment will be made, the manner in which information will be shared between parties and the legal ramifications of violating any commitment. TPAs can be both formal contracts and more informal commitments that do not carry the same legal weight. A TPA is not required to have all types of commitments defined to be valid.

Users can specify and manage the items covered by the agreement. Items can be specified as just purchased items, or as more commonly done in outsourced manufacturing, as top level assemblies and the specific components covered. The module 40 provides a number of mechanisms to specify these top-level assemblies and components. The module 40 also provides

a mechanism to group the items for assignment to specific agreement terms and conditions. The module 40 includes a library of pre-defined terms 18 that can be assembled into an agreement. Users can also create their own terms using the system's framework and functions for building terms. The terms contain quantifiable elements that are used as inputs to the liability calculation engine 50. Items on the agreement may be grouped together and assigned to agreement terms. Changes to agreements are managed through version control.

One of the differentiators of the present system over the prior art is its ability to combine agreements and supply chain data to calculate and present trading partner liability. The calculations may be a batch job performed by the liability calculation engine 50. The liability calculation engine 50 determines trading partner liability 22. Trading partner liability 22 is the results of all terms on the agreements being applied to the current supply disposition, including the valid and invalid attribution determined by the good faith calculation, to determine the true liability of each trading partner. Another element of the present invention is the good faith calculation, as performed by the good faith calculation engine 60. The good faith calculation engine 60 determines good faith liability 24. Good faith liability 24 is the results of the attribution of supply as being valid or invalid, according to the good faith calculation. The good faith calculation is described below in greater detail.

Figure 3 illustrates the inputs and outputs of a good faith calculation 300, which is performed by good faith calculation engine 60. The good faith calculation uses the historical demand signals from the customers, along with the current supply picture, to determine what level of supply the supplier should position if the supplier follows reasonable supply chain practices along with terms and conditions specified in the agreement. In one embodiment, the good faith calculation may utilize several standard transaction records from traditional ERP systems. Some of the records represent the current status of supply within the system. These may include records of exiting Purchase orders (POs) 302, Purchase Order Receipts (PO Receipts) 304 and Inventory positions 306. These records define how supply enters the system.

Other records from the ERP systems represent demand. These may include forecasts 308, existing Sales Orders 310 and order shipments and inventor pulls/consumption 312. These

records define how supply leaves the system.

The transaction data is evaluated in the context of the item definitions and the relationships between the items and bill of material (BOM) 314. The good faith calculation resolves the relationship between supply and demand for trading partners 316, and determines the resulting supply that is valid (needed to meet current and historically necessary demand) and invalid (supply acquired in excess of demand at the time it was acquired).

The output of the good faith calculation 300 is a determination of which Purchase Orders (POs) 318 and Inventory 320 have been acquired to meet necessary demand (i.e., valid) and which Purchase Orders (POs) 322 and Inventory 324 has been acquired in excess of demand (i.e., invalid).

Term Library

The following section provides a description of some exemplary terms that the system provides in one embodiment of the system's term library 18.

RIOH

RIOH (required inventory on hand) is a term that defines the level of buffer stock that a supplier agrees to maintain for a customer. RIOH can define the minimum level, the maximum level, or both the minimum and maximum levels of inventory. The levels can be specified as a fixed quantity or as a number of periods of demand. The present system allows the user to define periods of demand in a variety of ways that we refer to as demand calculation methods. The basic elements of a demand calculation method are the type of demand (a baseline demand, current forecast, sales orders, shipments), the direction of demand (forward looking or backward looking), the number of periods and the period size, and whether the RIOH is a sum of the number of periods or a sum of a number of periods based on an average of another number of periods. Here are some examples:

RIOHMAX-FCST: The Seller is required to maintain on hand inventory levels not to exceed quantities based on {N} {PERIODS} of current forecast.

RIOHMAX-SOHIST: The Seller is required to maintain on hand inventory levels not to exceed quantities based on {N} {PERIODS} of sales history.

RIOHMIN-FCSTAVG: The Seller is required to maintain on hand inventory levels not to exceed quantities based on {N} {PERIODS} of demand, where demand is defined using an average of {X} {PERIODS} of forward looking forecast.

RIOH can be used as an input to the determination of liability between two trading partners. For example, the buyer is liable for all inventory the seller maintains, up to the maximum specified in the RIOH term.

Time to Replenish

The time to "replenish" term may be used in conjunction with the RIOH term. It allows the user to specify how long it will take to replenish the buffer stock.

For example, in a Vendor Managed Inventory (VMI) replenishment environment the supplier carries a buffer inventory to supply product when demand spikes above what the forecasts predicted. Assume a CM (contract manufacturer) asks a CS (component supplier) to carry two weeks of buffer stock inventory based on the forecast baseline. This insures the CM can pull up to three weeks of product in one week (one week from the forecast and two weeks from the RIOH). Once the RIOH is depleted it will take a set amount of time to replenish the RIOH. The amount of time is called "time to replenish". The term indicates that it takes three weeks to replenish the inventory. That means the CS can supply the CM with one week of forecast but it will take three weeks to be able to provide the full buffer stock on demand.

Freshness Term

In VMI agreements, buyers require suppliers to hold a certain level of inventory on hand. Vendor Managed Inventory (VMI) is a type of relationship that exists between a customer and a vendor of material where the supplier of material is responsible for ensuring that an agreed upon level of inventory is stored at a location (frequently called a "hub") where the customer has

access. The customer "pulls" inventory as necessary and the vendor subsequently "replenishes" into the location. This type of agreement is referred to as VMI because it is the vendor's responsibility for ensuring that sufficient inventory exists, not the customer's. Part of VMI is the sharing of information where the customer has access to how much is currently located at the hub and the vendor has access to how much is pulled by the customer. In some cases, the "hub" is actually at the customer location and in others it is at an agreed upon third party location.

To contractually cover the liability with holding this inventory, suppliers include a term – "freshness," to ensure it is being pulled at an appropriate level. The objective of the freshness term is to ensure that the buyer is pulling that inventory. It typically comes into play to make sure the stock is rotated frequent enough not to become "stale." From a calculation perspective, the freshness term will be additive to the on-hand liability. The freshness term denotes when inventory becomes stale, and denotes the portion of stale inventory that the buyer is responsible for.

Term Description: Freshness - All inventory that has been in hub inventory longer than {N} {Periods} shall be considered past its freshness date.

Flexibility

"Flexibility" is a term that allows the buyer and supplier to specify the amount that demand can vary from a baseline. It helps to set expectations of reasonableness when buyers are forecasting their demand and dramatically increase or decrease the demand they had previously shared with the supplier. The term allows users to set up zones of time and a maximum percentage increase or decrease from the baseline demand that is allowed for that zone. A user can also set up overlapping flex limits called NTE (not to exceed). For example,

Flex up and not to exceed: The buyer can flex up 30% per month from a baseline but may not flex up more 15% per quarter.

The effect of the not to exceed limit means that although the buyer can flex up by 30% per month, once he has flexed to 15% above of the total quarter baseline, he cannot flex up in the

remaining months of the quarter.

The output of the flexibility term can be viewed in reports where buyer and supplier can see if the current demand exceeds the flexibility limits or how far they can go before they exceed the flexibility limits. Flexibility can also be used in determining liability. For example, the expedite term can be set up to apply an expedite premium to the whole quantity of an expedite or only the portion above the flexibility limit.

Reschedule Term

The "reschedule" term allows user a number of ways to define the limitation on rescheduling of an order. There are four basic ways that the user can define the reschedule limits.

1. The maximum number of times that an order can be rescheduled. The system allows a user a defined grace period where reschedules do not count toward this limit, after that the system counts each change of the delivery or ship date as a reschedule.
2. The general reschedule time fence. There is a reschedule in time fence and a reschedule out time fence. These time fences apply limits to all orders regardless of when they were placed or when they are due. For example, no orders can be rescheduled in less than 10 days from the current date, or no orders can be scheduled out beyond 90 days from the current date.
3. Order specific reschedule time fence. From its original due date, a maximum number of days in and a maximum number of days out can be specified.
4. Order based reschedule time fence within a zone. The user sets up zones similar to the flexibility and cancellation terms. The current due date determines which zones applies. Within each zone, the user specifies the maximum reschedule in and maximum reschedule out relative to the current due date. The user can also specify the percentage of the quantity of an order that can be rescheduled in or out.

The reschedule term can be specified as any one of the above four or any combination.

Cancellation Term

The "cancellation" term allows user to specify the penalties incurred when canceling an order. Similar to the flexibility and reschedule terms, users can set up zones of time and a cancellation percentage based on the order value and/or a cancellation fee (per order). For example, orders due between today and three weeks from today will incur a 100% cancellation cost, orders cancelled between four and eight weeks from today will incur a 50% cancellation cost and orders cancelled beyond eight weeks will incur a $50 cancellation fee per order.

The cancellation term can be used to compute liability between trading partners. For example, the buyer might be liable for all orders placed within the manufacturing lead-time of an item. The amount of liability is based on the order due date and the cancellation term.

Expedite

The "expedite" term allows the user to specify the premium that a buyer would pay if it were able to purchase products outside of established lead-time or flexibility limits. The term does not guarantee the availability of supply. It only defines the cost premium if supply is available. This is primarily used in what if scenarios where the potential demand exceeds lead-time or flexibility limits. Typically, the expedite orders are within the lead-time of the products. The supplier may still be able to supply the quantities above the limits but they will charge a higher price. The basic elements of the Expedite term are:

1. Expedite zones. The user can specify a schedule of zones where an expedite premium applies for each zone. The first zone can optionally be specified as a frozen zone where expedites are not allowed.
2. Expedite premium
    a. Percent of item cost or price *:
    The expedite premium can be set up for each zone in the expedite schedule as a percent of the item's cost or price.
    b. Fixed expedite charge per expedite *:
    The expedite premium can be set up for each zone in the expedite schedule as a fixed charge. The fixed charge is in addition to any other expedite charges such as percent

of cost premium or expedite cost per unit.

c. Cost or percent of cost at the item or item/vendor level *:

The cost schedule can be set up at the agreement level, the item level or the item/vendor level. The calculation engine will look for the expedite schedule in the following order of preference: i) item/vendor, ii) item level, and iii) agreement/item group level. If the higher preference schedule is not found it will look for the next priority schedule.

d. Flexibility Limits:

The user can optionally specify whether the expedite premiums should only apply to amounts outside of the flexibility limits. Schedule changes within the flexibility limits would not be charged a premium.

The system 10 may generate graphical user interfaces to allow users to view, enter and modify terms in the term library 18. Figure 19 is an example of a user interface 1900 that the system may generate to display the names and description of certain terms that may be present in an agreement.

Detailed Explanation of the Operation of the Good Faith Calculation Engine

Good Faith

The good faith calculation performed by the good faith calculation engine 60 ("Good Faith") characterizes the extended supply chain as a directed acyclic graph. The nodes are the trading partner sites. The links are the agreements with a specific notion of who the buyer (FROM partner) and seller (TO partner) are. Figures 4 and 5 show simple examples of extended supply chains. Figure 4 shows an OEM (Original Equipment Manufacturer) contracting with two CMs (Contract Manufacturers). Figure 5 shows a CM providing services to two OEMs.

The purpose of these relationships is to manufacture finished goods for sale. One party (OEM) defines the deliverable and demands from the supply chain some number of finished goods over time. The BOM (Bill Of Materials) defines the finished good to be provided. It contains a hierarchical specification of all parts (Items) contained in the finished goods. Figure 6

is an example of a simple BOM. It shows that the TLA (Top Level Assembly) is composed of three sub-assemblies, ASM1, ASM2 and ASM3. ASM1 is built out of two components, CMP1 and CMP2. ASM2 is built from CMP2 and CMP3. And ASM3 is built from CMP3 and CMP4. It is common to have the same item in multiple assemblies that are purchased by the buyer.

In the good faith calculation, demand can be passed from the buyer to the seller in many forms (e.g. forecast, sales orders, shipments/consumption). For the purposes of this document, the demand will be referred to as "the forecast." The forecast is the total demand that the buyer expects the seller to satisfy. If the buyer already has some supply, it has been accounted for ("netted") in the forecast. Each node in the supply chain is responsible for fulfilling the demand placed on the node (by the forecast) and will need to acquire supply for that fulfillment. A possible mapping of this BOM to the network in Figure 4 could be that the OEM is responsible for final assembly of the TLA. To do so it will need to acquire (demand) ASM1, ASM2 from CM1 and acquire (demand) ASM3 from CM2. CM1 and CM2 will then acquire (demand) CMP1, CMP2, CMP3 and CMP4 from the component suppliers.

The good faith calculation/engine understands the propagation of demand throughout the extended supply chain. At any instant in time, it applies the rules of a standard MRP (Materials Requirements Planning) system to the demand signals received by a site to create an expectation of what the site should do to satisfy the demand. This expectation is compared to the actual supply plan that the site has in place (purchase/production orders and inventory) to assess what is necessary at that instant in time to meet the demand. Good Faith maintains a historical record of the individual evaluations so that all supply can be tracked back to its origination and whether that supply should have been generated at the time. Various business decisions allow for the actual supply plan to deviate from minimum necessary for a number of reasons. For example, a CM could place a large order for a component to receive special pricing, with the expectation that the buyer will eventually consume all the supply. Good Faith tracks the fraction of that order that is required to meet the actual demand over time. If demand changes, an accurate accounting of exactly what portion of the order was necessary to meet the demand and what portion is still in excess of any actual demand is known. Good Faith assesses supply as "Valid" when at some time, T, in the past the supply was needed to meet the demand at time T. Good

Faith assesses supply as "Invalid" if, for the time between its origination and now, the supply has remained in excess of demand within lead-time for the item. Good Faith also understands the contractual terms that a buyer has with a seller. When there is excess supply that can be mitigated for no cost (e.g. buyer can cancel a purchase order without fees), then Good Faith "expects" the buyer to cancel the order and the relevant supply is considered "Invalid" since it can be removed from the system for no cost. Various rules can be applied when the cost is non-zero to assess when the buyer "should" cancel the order and the supply "should" be considered "Invalid".

Figure 7 shows the top level 700 of the good faith evaluation. It consists of an outer loop that processes each order site 702, as shown in block 704. Within each site, all the items pertinent to that site are evaluated for good faith execution.

Good Faith orders the sites such that all buyers are evaluated before any seller from which they purchase material. This is done by a conventional topological sort based on the dependencies specified by the agreements between the sites. For each site, the set of items that demand must be propagated is identified. These items may include three sets of items and all their components as specified in the BOM.

1. One set is the items that are purchased BY a site from this site. These items are Purchase-By Items 706. The calculation 700 gets these items in step 712.

2. A second set is the items that are "Manufactured" by this site or Manufactured Items 708. These are the items that this site builds from components ordered from other sites. The calculation 700 gets these items in step 714.

3. The third set is the items purchased by this site FROM another site. These are Purchase-From Items 710. The calculation 700 gets these items in step 716.

These sets of items as well as all of the children in the BOM 718 are the set of items for which this site must propagate demand, i.e., Order Items 720. The total set of Order Items 720 is sorted topologically based on the BOM hierarchy. When traversing the BOM to determine all

the children of the Purchase-By, Manufactured, and Purchased-From items (All_Items), are starting points for traversing the BOM in a depth first order. The BOM traversal terminates descent when the item is considered "Purchased" by the site being processed. Purchased items will have their demand "pulled" by the seller sites when they are processed, according to the processes shown in Figures 9A and 9B.

For each item in topological order, the following is performed (as shown by block 722):

1. Propagate the demand for the item (block 724), as shown in Figure 8 to determine a Propagated Demand 726.
2. Determine the set of agreements where the current site is the TO partner (seller) that cover the item (block 728). This determination provides an Item – TPA map 732. This map contains the set of agreements that the item needs to be evaluated on.
3. Evaluate the item with respect to the agreements and "Good Faith" practices (block 730), as shown in Figure 10.

Figure 22 is an example of a user interface 2200 that the system may generate to display the good faith results calculated for an item in a certain period.

Figure 8 shows a process 800 of propagating the demand for an item, which may be employed by the present invention. Good Faith must propagate demand across site boundaries. That is, Good Faith propagates a demand for an item at a buyer site when evaluating the buyer site. When the seller site is evaluated, the propagated demand acts as the "forecast" for assessing what the seller site is expected to do. The process for propagating demand for an item is:

1. Get the independent demand for the item in block 810 (Figures 9A, B). Demand 818 may be a forecast for an item loaded from a conventional Enterprise Resource Planning (ERP) system. In the preferred embodiment, the system supports consumption logic utilizing existing sales orders as well.
2. If the item is one of the Purchase-From, Manufactured, or Purchase-By items, the independent demand 814 is determined to be the propagated demand (in block 816).
3. For all other items, loop over all parents of the item in the BOM (block 820) and get the propagated demand for the parent (block 822).

4. Net the parent demand (block 824) with the parent inventory (block 826). Add the netted parent demand to the independent demand for the item (block 828). Summed demand 830 is the internal data structure for storing the demand for further processing. Block 832 is the control flow back to block 724 of Figure 7.

The propagated demand for the item is the sum of the independent demand and all of the dependent demand (propagated demand netted with inventory) for the parents. Figures 9A and 9B show a process of getting the Independent demand for an item, which may be employed by the present invention. Good Faith is designed to operate in a multi-enterprise environment. That is, ERP data from multiple sites provides the data that Good Faith must operate on. This includes the fact that the data relevant to a single item in the supply chain can require multiple identifiers to access the data. That is, the forecast for a Top Level Assembly may have been received from an extract of the purchaser's ERP system. In this case, the forecast will be in the purchaser's numbering scheme. Supply for the item will likely be held at the seller's site. The disposition of that supply (orders and inventory) may be received from an extract of the seller's ERP system. In this case the supply data will be in the seller's numbering scheme. Good Faith will correctly resolve the numbering scheme/data source issues. This problem is most pronounced with demand (forecast). To resolve some of the issues, Good Faith defines two directions for forecasts, incoming and outgoing. Incoming forecasts are defined to be forecasts that a site has received from a buyer site. The data is extracted from the seller's ERP system and is in the seller's number scheme. An outgoing forecast is one that has been "generated" by the site. It is a forecast of internal need or is to be sent to a supplier. It may be extracted from the "buyer's" ERP system. The separation of incoming and outgoing is needed since a site can be both a buyer and a seller in the system. Referring to Figures 9A and 9B, Good Faith may determine the Independent demand of an item by processes 900a, 900b as follows:

1. If the item is one that is purchased by another site (Purchased-By Items 902) from this site then:
2. Find all the sites that buy the item from this site (block 904) and sum the outgoing demand (block 906) from the database store (Demand 908).
3. If demand was found, the sum of that demand (Summed Demand 910) is the Independent

demand. This is returned in block 912.

4. Look for incoming demand from the database store (block 908) for the item.
5. If found, that is the independent demand for the item. This is returned in block 914.
6. If this item is purchased by this site from another site (Purchased-From Items 916) or this item is manufactured by this site (Manufactured Items 918) then:
7. Is there outgoing demand in the database store (Demand 922) from this site? If so, that is the Independent demand. This is returned in block 920.
8. For each site that sells the Purchased-From Item 916 to this site (block 924), sum all the incoming demand for the item at the selling site (block 928) producing Summed Demand 932.
9. If there is incoming demand at the seller sites, the sum is the independent demand for the item. This is returned in block 930.
10. If the item is an item purchased by another site (Purchased-By Items 936) from this site, look for propagated demand 942 at the buying site. This demand was generated when the buyer site was processed.
11. Find all the sites that buy the item from this site and sum the propagated demand (block 940).
12. If demand was found (Summed Demand 944), that is the independent demand. This is returned in block 946.
13. This item has zero Independent demand. This is returned in block 934.

Figure 10 shows a process 1000 for evaluating each item, which may be employed by the present invention. At a site, an item can be, and in general will be, on multiple agreements. Typically, there is one agreement covering the item where the site being processed is the TO site. That is the current site that is selling the item or assemblies containing the item. There are agreements with the suppliers FROM which this site purchases the item. Some of the agreement terms need to be evaluated before assessing the validity of the supply at the site. This is because some terms (e.g. Required Inventory On Hand, RIOH) place specific performance objectives on the site providing the supply. Other terms (e.g., liability) need to be evaluated after the assessment of the validity of supply at a site. This is because the buyer of an assembly will typically allow for the "positioning of supply in accordance with reasonable purchasing

practices." "Reasonable Purchasing Practices" should be understood as the practices expected to be utilized to minimize inventory needs. These may include but are not limited to following the recommendations of standard MRP (Material Requirements Planning) systems for material order placement and modification as well as return of excess material to vendors. Good Faith assesses what supply has been positioned in accordance with reasonable purchasing practices. The evaluation of an item proceeds as follows:

1. Evaluate the Pre-Good Faith terms 1002 covering the item on agreements where this current site is the TO site (blocks 1004, 1006). (See Figure 17). Pre-Good Faith terms are that are required to be evaluated before the updating of the good faith status of supply. In the preferred embodiment of the system, the term definition has a field attributing the term to be a Pre-Good Faith term.
2. Update the assessment of good faith status of the supply at the current site (block 1008). (See Figure 11).
3. Evaluate the remaining terms covering the item on agreements where this current site is the TO site (blocks 1010, 1012). (See Figure 17).

Liability terms are all terms that are associated with a Trading Partner Agreement (TPA) that are not Pre-Good Faith terms. These terms are evaluated after the good faith status of the supply has been updates. The results, Item TPA Results 1014, are stored in a database.

Figure 11 shows a process 1100 for tracking and updating the disposition of supply at a site with respect to good faith assessment of validity, which may be employed by the present invention. Arrival and consumption of inventory since the last assessment must be determined. Any discrepancies between expected values and actual values must be eliminated. Then the current demand must be compared with the current supply plan to determine which supply has been accumulated by "reasonable purchasing practices."

The updating of good faith status of supply may proceed as follows:

1. Determine receipts from Orders 1118 (in block 1102) (see Figure 12). The receipts became new inventory at the site. Orders 1118 may represent the purchase orders and

production orders from an ERP system specifying the anticipated receipt of additional supply, either through the delivery from a supplier in the case of purchase orders or the finalization of manufacturing in the case of production orders.

2. Determine the consumption 1120 of the inventory that has occurred since the last assessment (block 1104). Consumption 1120 may represent the amount of inventory that was used to either satisfy shipments in the case of finished goods or the amount of inventory pulled for use in the manufacture of a product.
3. In a perfect world the current inventory level would be equal to the old inventory level plus any receipts into inventory minus any consumption. However there are activities that take place outside the scope of the system that will cause discrepancies in the totals. The good faith status of the inventory is adjusted to match the current inventory level in block 1106 (see Figure 13). Inventory 1122 may represent the records from an ERP system specifying how many units of an item are on hand in a particular location.
4. Once the good faith status of the supply matches in quantity, then the demand for the item is used to establish the validity of the current supply. The current total of valid inventory and valid orders are subtracted from the demand. The net amount is the amount of additional supply that should be accumulated by reasonable purchasing practices (block 1108).
5. In block 1110, the process determines if the demand is less than or equal to zero. If it is, the process ends. If the net demand is greater than 0, promote invalid inventory to valid to meet the net demand (block 1112) (see Figure 14).
6. If after promoting all inventory there is still net demand remaining, promote invalid orders to valid (blocks 1114, 1116) (see Figure 15). Otherwise, the process ends.

Figure 12 shows a process 1200 for tracking changes to the status of orders for the item, which may be employed by the present invention. Any new supply (a new order or an increase to an existing order) is considered invalid until assessed against demand. Any decrease in supply (decrease in order quantity) reduces the amount of invalid and then valid quantities on the order. The process for determining new orders, order changes and new order receipts may be performed as follows:

1. For each order for the item (block 1202):
   a. If this is a new order (as determined in block 1204), create an order state to track the order and initialize the quantity on the order to invalid (as shown in block 1206).
   b. If the quantity on the order has increased (as determined in blocks 1208, 1210), increase the amount of invalid quantity on the order by the amount of the order increase (as shown in block 1212).
   c. If the quantity on the order has decreased (as determined in blocks 1208, 1210) reduce the amount of invalid quantity on the order by the amount of the decrease in the order (as shown in block 1214).
   d. If there were any receipts against this order (as determined in block 1216):
      i. Create an inventory state 1222 for the receipts (as shown in block 1220). Receipts from the order are considered to be the valid quantities and then the invalid quantities from the order. Inventory State 1222 is the data stored in the database reflecting the status of the inventory. Each receipt of inventory is tracked from inception through to consumption. For each receipt, the origin date, original quantity, the remaining valid units, remaining invalid units and the date of completion if totally consumed may be stored.
      ii. Reduce the quantities on the order state 1224 by the amount received (as shown in block 1218). Order state 1224 is the data stored in the database reflecting the status of the order. In the preferred embodiment, orders can be either purchase orders or production orders. Each order is tracked from inception through to completion. For each order, the origin date, original quantity, the remaining valid units, remaining invalid units and the date of completion if completed may be stored.

Figure 13 shows a process 1300 that may be employed by the present invention for adjusting the Good Faith status of the inventory to match the quantity actually loaded from the ERP system. Various events can occur (e.g. cycle count) that can cause adjustments to the inventory total without transaction records (pulls or receipts). The process 1300 may be performed as follows:

1. Take the consumption and net it against the current inventory states (block 1302) (See Figure 11).
2. If there is consumption remaining after it has been netted against the all the inventory as of the last assessment with all the receipts received since the last assessment (as determined in block 1304), then note the excess consumption (as shown in block 1306).
3. Compare the inventory status with the total from the ERP system (as shown in block 1308). If the expected inventory level is greater than the quantity from the ERP system, consume the excess inventory (as shown in blocks 1316, 1318).
4. Compare the inventory status with the total from the ERP system. If the ERP total is greater, add additional invalid inventory to make up the difference (as shown in blocks 1312, 1314).

Figure 14 shows a process 1400 for determining which inventory is consumed. First, all valid inventory is netted with the consumption. As the inventory is tracked by when it was received, the inventory received earliest is consumed first. If after consuming all valid inventory there is consumption remaining, consume invalid inventory again by receipt date. The process may be performed as follows:

1. Order all inventory by receipt date.
2. For each bucket of inventory in receipt date order:
    a. If there is no valid quantity in this bucket, proceed to the next (blocks 1402, 1404).
    b. If the valid quantity is greater than or equal to the remaining consumption, reduce the valid quantity by the consumption and set the remaining consumption to 0 (blocks 1406, 1408).
    c. If the valid quantity is less remaining consumption, set the valid quantity to 0 and reduce the remaining consumption by the amount of valid inventory (blocks 1406, 1410).
    d. If there is consumption remaining, then for each bucket of inventory in receipt date order (block 1412):
    e. If there is no invalid quantity in this bucket, proceed to the next (block 1414).
    f. If the invalid quantity is greater than or equal to the remaining consumption,

reduce the invalid quantity by the consumption and set the remaining consumption to 0 (blocks 1416, 1418).

g. If the invalid quantity is less remaining consumption, set the invalid quantity to 0 and reduce the remaining consumption by the amount of invalid inventory (block 1420).

Figure 15 shows a process 1500 for promoting inventory from invalid to valid, which may be employed by the present invention. The amount to promote is determined by taking the demand for the item in excess of the total valid supply (inventory and orders) and promoting invalid inventory until there is no more inventory to promote or sufficient inventory has been promoted to have the total valid supply match the demand for the item. The process 1500 may be performed as follows:

1. Order all inventory by receipt date.
2. For each bucket of inventory in receipt date order (block 1502):
    a. If there is no invalid quantity in this bucket, proceed to the next (block 1504).
    b. If the invalid quantity is greater than or equal to the remaining demand, reduce the invalid quantity by the demand, increase the valid quantity by the demand and set the remaining demand to 0 (blocks 1506, 1508).
    c. If the invalid quantity is less remaining demand, increase the valid quantity by the amount of invalid quantity, set the invalid quantity to 0, and reduce the remaining demand by the amount of invalid inventory (blocks 1506, 1510).

Figure 16 shows a process 1600 for promoting orders from invalid to valid, which may be employed by the present invention. The amount to promote is determined by taking the remaining demand for the item in excess of the total valid supply (inventory and orders) and promoting invalid orders until there is no more order quantity to promote or sufficient order quantity has been promoted to have the total valid supply match the demand for the item. When promoting orders, the order modifiers specific to the order are accounted for by promoting the first quantity on a specific order by the minimum order quantity. Subsequent promotions of order quantity on that specific order are then in order multiple quantities. The process 1600 may be performed as follows:

1. Order all orders by due date.
2. For each order in due date order (block 1602):
    a. If there is no invalid quantity in this order, proceed to the next (block 1604).
    b. If the invalid quantity is greater than or equal to the remaining demand, reduce the invalid quantity by the demand, increase the valid quantity by the demand and set the remaining demand to 0 (blocks 1606, 1608).
    c. If the invalid quantity is less remaining demand, increase the valid quantity by the amount of invalid quantity, set the invalid quantity to 0, and reduce the remaining demand by the amount of invalid order quantity (blocks 1606, 1610).

Figure 17 shows a process 1700 for calculating the liability terms, which may be employed by the present invention. The process 1700 may be performed as follows:

1. For each term 1714 to be evaluated (block 1702)(terms 1714 may comprise Pre-Good Fait terms 1002 and/or Liability terms 1014):
    a. Get the term definition (block 1704). Term definitions 1710 may be stored in the database. The definitions may include equations, expressed in the calculation functions described below, and results to be calculated for the term.
    b. Get the parameters defined for the term (block 1706). Term parameters 1712 can qualify behavior of the term. For instance, there may be a RIOH term that specifies that N periods of supply are to be maintained. In one instance, that might specify N=2 and periods = weeks, and another might specify N=1 and periods = month.
    c. Evaluate the term by invoking an interpreter on the equation for each expression and storing the value of the expression as a result (block 1708).

Figure 21 is an example of a user interface 2100 that the system may generate to display liability that exists between a company and a customer due to the terms included on an agreement and the orders and inventory that exist in the relationship.

How The System Performs Calculations

Calculations are performed in the context of an item. The item provides the reference point for determining the actual data sets to use in the calculations. The data sets are operated on by the functions that make up the operational language for the calculations.

The liability terms are expressed in the system as a series of expressions with a set of parameters. Each expression is a result and an equation. The result is a string containing a label for identifying the value. The equation is a string containing a statement made from language primitives. The statement is evaluated by parsing and executing the language primitives. This may be accomplished with a standard interpreter implementation with the language primitives implemented by semantic routines. The result of the evaluation is assigned to the symbol defined by the result. The result can then be referenced in subsequent equations. The language primitives can be extended by adding new functions that follow a defined interface.

There may be two forms for the results that are calculated. One simple form is a scalar. It represents a single value. A second form is a Time Series. A Time Series is a sequence of non-overlapping time ranges or buckets. Each bucket has a value associated with it, though the value can be null. The buckets are effectively always contiguous since if the Time Series has buckets that are non-contiguous, it is interpreted as having a bucket with a null value covering the range between the two non-contiguous buckets. The Time Series is a powerful abstract data type that manages all the details of operating on time sequenced data with differing time bases.

When operations are performed on Time Series, the time buckets must be aligned. This is accomplished by subdividing buckets so all the time ranges are same. The values of the new buckets are determined by taking the fraction of the new bucket compared to the original bucket and making the new bucket value the same fraction of the original bucket value. Time Series understand the calendar when aligning the two series so that dividing of buckets accounts for workdays versus off days. Figure 18 depicts one example of the results of aligning two Time Series. Once the two series are aligned, the math operations may be performed on a bucket-by-bucket basis.

The expressibility of the calculation engine comes from the ability to combine terms through a simple syntax. The basic math operations of addition, subtraction, multiplication and division (+,-,*,/) are supported as well as absolute value (abs) and taking the negative (negate). The language for expressing the liability terms provides terms for accessing ERP data as well as manipulating specific data sets. The system defines some special results that the liability terms for each item must define if there is to be liability for the item. In one embodiment, the symbols include:

AOO – Actual Items On Order
AOH – Actual Items On Hand
OO – Items on Order that the buyer is liable for
OH – Items on Hand that the buyer is liable for
AOOCOST – Monetary valuation of the actual Items on Order
AOHCOST – Monetary valuation of the actual Items on Hand
OOCOST – Monetary valuation of the Items on Order that the buyer has liability for
OHCOST – Monetary valuation of the items on Hand that the buyer has liability for

Functions related to Items

Many of the functions are directly related to the item that defines the context in which the functions are being invoked. Each item has, in general, two identifiers in the system because of the two parties involved. The system provides general access to these two identifiers through the defined symbols FROMITEM and TOITEM. These are the identifiers for the item as known by the "From" trading partner and "To" trading partner of the TPA.

The functions that are related to items have a default identifier (either FROMITEM or TOITEM) that is used to query the database for specific records. The default can be over-ridden by adding the desired identifier as the first parameter of the function invocation.

In addition to the symbols for identifiers, there are two other predefined symbols that are available to any calculation. They are TODAY which is set to the current System Data Date and LEADTIME which is set to the fixed lead time in days for the item as specified in the item data.

The Calculation Functions

In the preferred embodiment, the system provides the following calculation functions:

BASELINE

The "Baseline" or BASELINE calculation gets the baseline for an item using a rolling forecast model

BASELINE()

CANCELLATION

The "determine the cancellation value" or CANCELLATION calculation applies the cancellation factors to determine the cost of canceling open purchase orders. A string parameter can be specified that determines the date field in the PO record that is used to collate the results. Legitimate values, in priority order, are DUEDATE, ORDERDATE, PROMISEDATE, REQUESTDATE. If no string parameter is specified that matches the date fields, the first non-null date in the priority order is used. Two date parameters can also be specified. The first date is the "fromDate" and only result dates after that date will be returned. The second date is the "toDate" and if specified, only dates before that date will be returned. The cancellation value is based on the order quantity minus the received quantity. If using "lead time" as the period code, then the start and end period values specified in the Cancellation Model define zones based on percentages of the lead time. If applying charges only outside of defined flexibility zones, then the appropriate charges will be calculated only for any open purchase orders that fall below the flex down limit

CANCELLATION("DUEDATE", fromDate, toDate)

COST

The "determine cost" or COST calculation applies the cost factor for the item to the parameter. The parameter is the result of a previous calculation that has the number of items.

COST(units)

FC

The "forecast" or FC calculation gets the propagated forecast for the item. If an item is a component of multiple assemblies, the forecast for the item is a combination of the forecasts for the assemblies based on the number of the items in the assemblies.

Up to three dates can be specified in the parameter list. The first date is the as of date. The default is to use the system current data date, but previous forecasts can also be referenced with the appropriate as of date. The second date is from date and if specified then buckets with a date before the From date will not be returned. The third date is the to date and if specified then buckets with a date after the to date will not be returned.

FC (asOfDate, fromDate, toDate)

GFOO

The "good faith on order" or GFOO calculation gets the quantity of orders that have been assessed by the "Good Faith" process to be valid.

GFOO ()

GFINV

The "good faith inventory" or GFINV calculation gets the quantity of inventory that has been assessed by the "Good Faith" process to be valid.

GFINV ()

INV

The "inventory" or INV calculation gets the inventory for the item. Up to two dates can be specified as parameters. If no parameters are specified then the inventory as of the system current data date is returned. If on date is specified then the inventory as of that date is returned.

If two dates are specified then the series of inventory values is return for inventory records between the From and To dates.

INV (from, to)

PO

The "Purchase Orders" or PO calculation gets the purchase orders for an item. A string parameter can be specified that determines the date field in the PO record that is used to collate the results. Legitimate values, in priority order, are DUEDATE, ORDERDATE, PROMISEDATE, REQUESTDATE. If no string parameter is specified that matches the date fields, the first non-null date in the priority order is used. Two date parameters can also be specified. The first date is the "fromDate" and only result dates after that date will be returned. The second date is the "toDate" and if specified, only dates before that date will be returned. Additional string parameters can be specified that match the status codes for the purchase orders. If either of the strings "OPEN" or "CLOSED" is specified, then only those purchase orders with statuses matching one of the strings will be incorporated in the results. A string parameter can be specified that determines the quantity field in the PO record that is used in the results. Legitimate values, in priority order, are ORDERQTY, RECEIVEDQTY, INSPECTIONQTY, RETURNEDQTY. If no string parameter is specified that matches the quantity fields, ORDERQTY is used.

PO ("DUEDATE", fromDate, toDate, "OPEN", "CLOSED", "ORDERQTY")

PRICE

The "determine price" or PRICE calculation applies the price factor for the item to the parameter. The parameter is the result of a previous calculation that has the number of items.

PRICE (units)

PRODORDER

The "Production Order" or PRODORDER calculation gets the production orders for the

item. A string parameter can be specified that determines the date field in the PRODORDER record that is used to collate the results. Legitimate values, in priority order, are DUEDATE, STARTDATE. If no string parameter is specified that matches the date fields, the first non-null date in the priority order is used. Two date parameters can also be specified. The first date is the fromDate and only result dates after that date will be returned. The second date is the toDate and if specified, only dates before that date will be returned. Additional string parameters can be specified that match the status codes for the purchase orders. If any of the strings "OPEN", "CLOSED", "PENDING" or "IN PROCESS" are specified, then only those production orders with statuses matching one of the strings will be incorporated in the results. A string parameter can be specified that determines the quantity field in the PRODORDER record that is used in the results. Legitimate values, in priority order, are ORDERQTY, COMPLETEDQTY, SCRAPPEDQTY. If no string parameter is specified that matches the quantity fields, ORDERQTY is used.

PRODORDER ("DUEDATE", fromDate, toDate, "OPEN", "ORDERQTY")

RETURN

The "stock return" or RETURN calculation computes the cost of returning inventory for an item. An optional argument is the string "cost" or "price"; the default is to use the cost of the item in computing the result. The next argument is a decimal fixed cost per item. The final argument is a decimal variable cost per item, expressed as an integer between zero and 100 signifying the percentage of the cost (or, optionally, price) of the item itself as computed by the normal COST (or PRICE) function.

RETURN (FIXEDFEE, VARIABLEFEE)

The formula is: inventory * (FIXEDFEE + costorprice * (VARIABLEFEE / 100)).

In the absence of that override, the function must determine how much of the total inventory is attributable to the current TPA. To do this, it sorts in reverse order by due date all purchase orders for the item which are explicitly in the "closed" state, or which are assumed to

be closed because they disappeared between one data-load operation and the next. It then traverses the list, pretending to return inventory to the vendor of each PO in turn, until it has depleted the total inventory. If the vendor and site of a PO disagree with the "to" form of the item, then the result computation ignores the part of the inventory attributed to that PO.

SO

The "Sales Orders" or SO calculation gets the sales orders for the item. A string parameter can be specified that determines the date field in the SO record that is used to collate the results. Legitimate values, in priority order, are REQUIREDDATE, ORDERDATE, PROMISEDATE, REQUESTDATE, SHIPMENTDATE. If no string parameter is specified that matches the date fields, the first non-null date in the priority order is used. Two date parameters can also be specified. The first date is the fromDate and only result dates after that date will be returned. The second date is the toDate and if specified, only dates before that date will be returned. Additional string parameters can be specified that match the status codes for the purchase orders. If any of the strings "OPEN", "CLOSED" or "PENDING" are specified, then only those sales orders with statuses matching one of the strings will be incorporated in the results. A string parameter can be specified that determines the quantity field in the SO record that is used in the results. Legitimate values, in priority order, are ORDERQTY, SHIPPEDQTY, ALLOCATEDQTY. If no string parameter is specified that matches the quantity fields, ORDERQTY is used.

SO ("REQUIREDDATE", fromDate, toDate, "OPEN", "ORDERQTY")

Functions not related to items

AVG

The "Average" or AVG calculation returns the average value of a series. This function takes three parameters. First is the series to be averaged. Second is the period code for the averaging buckets. Legitimate values are "DAY', "WEEK", "MONTH", "QUARTER". The third parameter is the number of periods. The first numPeriod periods of the series are averaged for the result.

AVG (series, period, numPeriods)

CALENDAR

The "calendar dates" or CALENDAR calculation gets specific dates relative to the current calendar. The calendar is the calendar for the From trading partner of the TPA.

Parameters:

Day – Date that the returned date will be relative to. Default is TODAY.

Specifier – String that specifies which date is desired.

"QUARTERFIRST" – Get the first day of the quarter containing the Day.

"QUARTERLAST"– Get the last day of the quarter containing the Day.

"MONTHFIRST"– Get the first day of the month containing the Day.

"MONTHLAST"– Get the last day of the month containing the Day.

"WEEKFIRST"– Get the first day of the week containing the Day.

"WEEKLAST"– Get the last day of the week containing the Day.

CALENDARADD

The "calendar math" or CALENDARADD calculation performs calendar arithmetic by adding the number of periods of the specified type to the given date.

CALENDARADD (day, numPeriods, periodType)

Parameters:

day – base day to add numPeriods of periodType to.

numPeriods – number of periods. Negative is OK

periodType – "DAY", "WEEK", "MONTH", "QUARTER"

CAT

The "concatenate the two series" calculation returns a single series that is the result of appending the second series to the end of the first.

CAT (series1, series2)

CHOP

The "remove unwanted portions" or CHOP calculation returns the portion of the series that is between the From and To dates.

CHOP(series, fromDate, toDate)

CONSUME

The "consume from the total of a series" or CONSUME calculation returns a series that contains 0 in the original buckets until the sum of the original values in the buckets exceeds amount. The last bucket value is set so the sum of the all the values removed from the series equals amount and the remaining values are the same as for the original series.

CONSUME(series, amount)

FIRST

The "first date" or FIRST calculation returns the first date of the series parameter.

FIRST(series)

FLEX

The "Calculate Flexibility" calculation or FLEX gets the contractual flexibility for the given forecast series. This function takes two parameters. First is the series that is the forecast to be flexed. Second is the direction to be flexed, either "UP" or "DOWN"

FLEX(forecast, direction)

A "Flexibility Model" – Must be defined for the term including the FLEX calculation, this will cause the FLEX calculation to calculate using a flexibility model as specified by the parameters in the Flexibility Model.

LIMIT

The "limit the total of a series" calculation or LIMIT returns a series that contains the original values of the series parameter until the sum of those values exceeds amount. The last value is set so the sum of the result series equals amount and the remaining values are 0.

LIMIT(series, amount)

LAST

The "last date" or LAST calculation returns the last date of the series parameter.

LAST(series)

MAX

The "maximum" or MAX calculation returns the maximum of two values

MAX(value1, value2)

MIN

The "minimum" or MIN calculation returns the minimum of two values

MIN(value1, value2)

NTE

The "Calculate Not To Exceed" or NTE calculation gets the contractual not to exceed for the item. The default identifier is the FROMITEM. This function takes three parameters. First is the series that is the forecast to be flexed. Second is the series that is the demand that has consumed part of the forecast. Third is the direction to be flexed, either "UP" or "DOWN"

NTE(forecast, demand, direction)

A "Flexibility Model" must be defined for the term including the NTE calculation, this will cause the NTE calculation to calculate using a flexibility model as specified by the parameters in the Flexibility Model.

REBUCKET

The "Rebucket" or REBUCKET calculation resets a series to known buckets starting at a known point. This function takes three parameters. First is the series to be rebucketed. Second is the period code for the new bucket times. Legitimate values are "DAY', "WEEK", "MONTH", "QUARTER". The third parameter is the number of period codes to use to make the new bucket size.

REBUCKET(series, period, numPeriods)

RSUM

The "running sum" or RSUM calculation returns a series where each bucket is the sum of all the buckets in the series parameter up to that point.

RSUM(series)

SUM

The "sums all values" or SUM calculation returns the total of all the buckets in the series parameter.

SUM(series)

TSUM

The "trailing sum" or TSUM calculation returns a series where each bucket is the sum of all the buckets in the series parameter beyond that point.

TSUM(series)

The foregoing calculations and parameters can be associated with any term. The term can be stored as part of the term library 18 and used with any agreement. Figure 20 is an example of a user interface that the system may generate to display parameters and calculations that are associated with a term (e.g., a Liability term).

Figure 23 is an example of a user interface 2300 that the system may generate to display the calculated inventory and order levels for a specific item base on the agreement terms and the forecasts, order and inventory that exist for that item.

While the invention has been particularly shown and described with respect to illustrative and preferred embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and details may be made therein without departing from the spirit and scope of the invention.